In [1]:

```
# We are using a submodule of scikit learn called tree.
#Decision Tree(Machine Learning Model)
from sklearn import tree
```

In [5]:

```
# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],[150,70,9],[90,55,3]]

Y=['male','female','female','male','male','female','male','female','male','male','femal
e']     # Gender of 11 persons of planet earth
clf=tree.DecisionTreeClassifier()   #clf variable stands for classifier
clf=clf.fit(X,Y)

prediction=clf.predict([[190,90,10],[160,45,5],[250,150,12]])  #we can pass here as man
y persons name as we want to...

print(prediction)
```

```
['male' 'female' 'male']
```

In [10]:

```
# Using nearest centroid classifier in scikit learn

from sklearn.neighbors.nearest_centroid import NearestCentroid

clf= NearestCentroid()

# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],\
   [150,70,9],[90,55,3]]

# Gender of 11 persons of planet earth
Y=['male','female','female','male','male','female','male','female','male','male','femal
e']



clf=clf.fit(X,Y)

prediction=clf.predict([[200,85,11]])

print(prediction)
```

```
['male']
```

In [12]:

```
# Using Random Forest Classifier in scikit learn implementing on the same data

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators=2)

# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],\
   [150,70,9],[90,55,3]]

# Gender of 11 persons of planet earth
Y=['male','female','female','male','male','female','male','female','male','male','femal
e']

clf = clf.fit(X,Y)

prediction = clf.predict([[90,55,3]])

print(prediction)
```

```
['female']
```

In [18]:

```
# Using Support Vector Machines (SVM) in scikit learn
from sklearn import svm

clf= svm.SVC(gamma='scale')

# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],\
   [150,70,9],[90,55,3]]

# Gender of 11 persons of planet earth
Y=['male','female','female','male','male','female','male','female','male','male','femal
e']

clf= clf.fit(X,Y)

prediction= clf.predict([[140,60,7]])

print(prediction)
```

```
['female']
```

In [27]:

```
# Using naive_bayes classifier in scikit learn
from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()

# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],\
   [150,70,9],[90,55,3]]

# Gender of 11 persons of planet earth
Y=['male','female','female','male','male','female','male','female','male','male','femal
e']

clf = clf.fit(X,Y)

prediction= clf.predict([[150,70,9]])

print(prediction)
```

```
['male']
```

In [28]:

```
# Using LogisticRegression in scikit learn
from sklearn.linear_model import LogisticRegression

clf= LogisticRegression()

# height , weight, shoesize of 11 persons
X=[[180,70,8],[160,45,5],[120,60,5],[190,90,10],[200,85,11],[100,60,5],[250,150,12],[12
0,55,6],[140,60,7],\
   [150,70,9],[90,55,3]]

# Gender of 11 persons of planet earth
Y=['male','female','female','male','male','female','male','female','male','male','femal
e']

clf= clf.fit(X,Y)

prediction = clf.predict([[120,55,6]])

print(prediction)
```

```
D:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:433: Future
Warning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv
er to silence this warning.
  FutureWarning)

['male']
```

In [29]:

```
# Using KNeighborsClassifier in scikit learn
from sklearn.neighbors import KNeighborsClassifier


clf = KNeighborsClassifier(n_neighbors=3)

#[height, weight, shoe_size]
X = [[181, 80, 44], [177, 70, 43], [160, 60, 38], [154, 54, 37], [166, 65, 40], [190, 9
0, 47], [175, 64, 39],
     [177, 70, 40], [159, 55, 37], [171, 75, 42], [181, 85, 43]]

Y = ['male', 'male', 'female', 'female', 'male', 'male', 'female', 'female', 'female',
'male', 'male']

clf.fit(X, Y)

prediction = clf.predict([[190, 70, 43]])

print (prediction)
```

```
['male']
```